

AD-A099 786

ROYAL SIGNALS AND RADAR ESTABLISHMENT MALVERN (ENGLAND) F/6 9/2
AN EXPERIMENTAL IMPLEMENTATION OF ATLAS PROGRAMS ON AN IEEE-488—ETC(U)

OCT 80 D P TAYLOR

RSRE-MEMO-3284

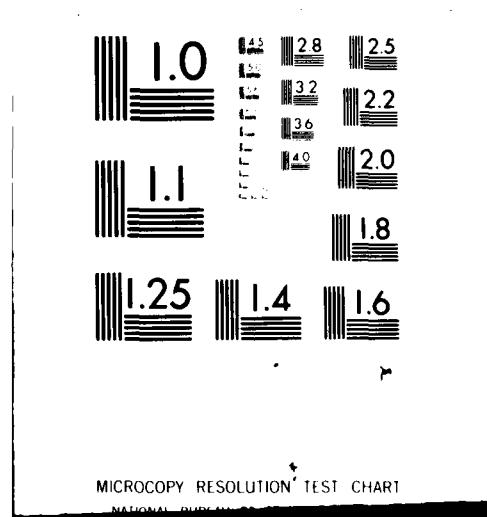
DRIC-BR-77373

NL

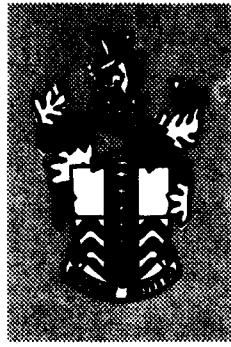
UNCLASSIFIED

1-6
200-1799

END
DATE FILMED
8-81
DTIC



AD A 09 9786



UNLIMITED

BR77575

(1)

LEVEL II

RSRE
MEMORANDUM No. 3284

ROYAL SIGNALS & RADAR ESTABLISHMENT

AN EXPERIMENTAL IMPLEMENTATION OF ATLAS PROGRAMS ON
AN IEEE-488 BUS STRUCTURED ATE

Author: D P Taylor

DTIC
ELECTE
S JUN 05 1981
D
E

PROCUREMENT EXECUTIVE,
MINISTRY OF DEFENCE,
RSRE MALVERN,
WORCS.

~~This document is the property of Procurement Executive, Ministry of Defence.
Its contents should not be made public without the express permission of the Director, RSRE.~~

RSRE MEMORANDUM No. 3284

UNLIMITED

(18) DRIIZ

(19) BR-77373

ROYAL SIGNALS AND RADAR ESTABLISHMENT

Memorandum 3284

(B) 13

Title: (6) AN EXPERIMENTAL IMPLEMENTATION OF ATLAS PROGRAMS ON
AN IEEE-488 BUS STRUCTURED ATE

Author: (10) D. R. Taylor

Date: (11) October 1980

(14) R-SPE-MEMO-3124

SUMMARY

The Memorandum describes the results of research work which has investigated, in outline, one technique to implement ATLAS programs on an experimental IEEE-488 bus structured ATE.

The work was performed to determine if it is feasible to combine two powerful ATE standards, the IEEE-488 bus and ATLAS, using an ATLAS pre-compiler resident on a PDP 11 and a simple ATLAS post-compiler resident on an Intel microprocessor development system. Although the extent of the work was limited the results were fairly encouraging.

This memorandum is for advance information. It is not necessarily to be regarded as a final or official statement by Procurement Executive, Ministry of Defence

Copyright
C
Controller HMSO London
1980

(A)

RECORDED

RSRE MEMORANDUM 3284

AN EXPERIMENTAL IMPLEMENTATION OF ATLAS PROGRAMS ON AN IEEE-488 BUS STRUCTURED ATE

D P Taylor

LIST OF CONTENTS

- 1 INTRODUCTION
- 2 INTRODUCTION TO THE IEEE-488 BUS
 - 2.1 Bus Characteristics
- 3 INTRODUCTION TO ATLAS
 - 3.1 ATLAS Characteristics
- 4 THE RESEARCH WORK
 - 4.1 Outline Description of the Software Aspects of the Research Work
 - 4.2 An Outline Description of the Hardware Aspects of the Research Work
- 5 CONCLUSIONS
- 6 REFERENCES

APPENDIX A

1 INTRODUCTION

This memorandum describes the results of research work which was performed to determine if it is feasible, by the use of the host computer to target Automatic Test Equipment (ATE) concept, to combine two powerful ATE standards, the IEEE-488 bus¹ and the Abbreviated Test Language for All Systems (ATLAS)². The attendant advantages of such a scheme being the ability to be able to program an ATE in a test orientated, high-level language, the program itself forming the conventional test specification, coupled with the flexibility of the target ATE configuration by virtue of the use of the IEEE-488 bus. For the work the host computer used was a multi-user PDP 11/40, and it was on this that ATLAS programs were compiled into an intermediate code. The intermediate code was then physically transported to an INTEL Microprocessor Development System (MDS) where it was translated, via a simple translator written in Pascal³, into a pseudo IEEE-488 bus language. The pseudo language consisting of macro calls, the macro definitions being in the form of CORAL 66 with assembly code inserts. The pseudo language thus formed the basis of a CORAL 66 program which was compiled on the MDS. At an early stage in the work it was decided to use the Modular Approach to Software Construction and Test (MASCOT)⁴ philosophy and this was achieved using a MDS based MASCOT supervisor and kernel. When test program verification had been performed, using the MDS, the resultant object code was transported to a minimum configuration target ATE.

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification _____	
By _____	
Distribution/ _____	
Availability Codes _____	
Dist	Avail and/or Special
A	

This paper does not attempt to describe in detail either the IEEE-488 bus or ATLAS as both standards are well defined. However to enable the description of the aims and results of the research work to be meaningful, to those not familiar with both standards, a preliminary introduction is given.

2 INTRODUCTION TO THE IEEE-488 BUS

The IEEE-488 bus, hereafter referred to simply as the bus, was first defined as an IEEE standard in 1975 which was amended in 1978 to bring it into line with the IEC TC66 standard in all respects other than the physical bus to instrument connector. The bus also goes under the names of General Purpose Interface Bus (GPIB), Hewlett Packard Interface Bus (HPIB) and ANSI MC1.1. Certain problems can occur when inter-connecting instruments with claimed bus capability but in general they can be overcome without too much difficulty. The bus was originally intended as a means of automating simple bench top instrumentation measurement functions, but over the past two years interest has been shown in using the bus as a basis for deriving ATE systems.

One very striking aspect of the bus is its popularity, most instrument manufacturers are now providing the bus as an optional extra and bus controlled switching matrices are available, as are fibre optic links which can be used to extend the physical length of the bus and also provide noise immunity. Bus interface chip sets are available for a number of popular microprocessors and complete bus controllers are available with the programming language Extended BASIC as a popular choice.

A few of the reasons for the popularity of the bus are:

- i. Because the bus provides a defined usable standard in terms of mechanical, electrical and device to device interface message characteristics, an ATE system designer knows precisely what parameters he is dealing with and consequently ATE development time can be reduced. Compare this with the earlier use of non-standard Binary Coded Decimal (BCD) device interfaces and the advantage is obvious.
- ii. The definition of the bus enables devices to be interconnected in any convenient way, for example linear or star, this means that the large number of cables and attendant interference/connection problems associated with BCD interfaces are considerably reduced.
- iii. The instrument fit and hence capability of a bus ATE can be changed with, in general, a minimum of software effort.

2.1 Bus Characteristics

The bus consists of sixteen parallel lines; eight for data, three for performing data handshakes and five bus management lines. In a bus system there can only be one active controller, generally the device which has computing power and there can be up to fourteen other devices, generally these will be instruments.

Devices other than the controller are categorised according to their capabilities. For example a Digital Voltmeter (DVM) will be required to receive set up instructions and then transmit the measured data which in bus terminology is a listener/talker device. Other devices, such as a lineprinter will only be listeners as will some devices only be talkers. In all but the bus configuration without a controller, each device is

assigned a unique address so as to enable device selection. Clearly, at any one time, there can only be one active talker although there may be a number of active listeners. The data bus is eight bits wide and for all operations data streams are handshaked from device to device on an eight bit parallel byte serial basis at a maximum data rate of one megabyte per second. This data rate is reduced to two hundred and fifty kilobytes per second if the maximum bus path length of 20 metres is reached.

3 INTRODUCTION TO ATLAS

ATLAS was originally developed for avionics applications under the auspices of Aeronautical Radio Inc (ARINC) and the first version was approved in 1968. The purpose of ATLAS is to provide a standardised test language for expressing test specifications and procedures which are independent of the target test equipment. In 1976, when ATLAS was approved as an IEEE standard, thirteen language revisions had occurred and the reprint of the ARINC revision, known as 13A was given the IEEE standard 416-1976. It was mentioned that the one striking aspect is its distinct lack of popularity, although most bodies involved with ATE all agree that considerable benefits to manufacturers and users alike can accrue from the use of a standard high-level test orientated language. To gain an appreciation of the reasons for ATLAS's lack of popularity consider the following. The ATLAS language controlling body consists of voluntary experts. There has throughout the history of the language been a remarkable lack of main purpose, in particular this being whether ATLAS is an ATE programming language or simply a test specification language. Because to make such a decision by voluntary committee is difficult and because few computer language experts have had any involvement in the early language definition stage a vague compromise position has been reached. In other words ATLAS has not been devised to simplify the problems of converting (compiling) ATLAS programs into ATE machine code. Put another way the job of deriving ATLAS compilers has, to date, proved to be too complex and expensive to encourage ATE manufacturers to supply an ATLAS compiler as part of their product range. Therefore it is not surprising that if ATLAS programs cannot be compiled the inducement to write in ATLAS at all is not commercially attractive.

A further glaring problem with ATLAS is its deficiency as a digital test language, although considerable efforts are being made at the present time to improve the situation.

3.1 ATLAS Characteristics

ATLAS is a rigorously defined test problem language orientated towards the Unit Under Test (UUT) rather than the target test equipment. A typical ATLAS statement with preceding commentary is of the form:

C MEASURE THE RESISTANCE BETWEEN UNIT UNDER TEST PINS J-1 AND J-2.
 THE MAXIMUM RESISTANCE EXPECTED IS 10 KOHMS \$.

000105 MEASURE, (RES), IMPEDANCE, RES MAX 10 KOHM,
 CNX HI J-1 LO J-2 \$

The language has many of the features associated with conventional problem orientated languages such as macro, conditional statement, block structuring and goto facilities. It does however have one major difference and that is the large number of language reserved words and hence syntax. To illustrate this point ATLAS has in the order of ten times more reserved words than CORAL 66.

4 THE RESEARCH WORK

The basic purpose of the research was to determine if it is feasible, by the use of the host computer to target ATE concept, to combine the two powerful ATE standards described, the bus and ATLAS. A subsidiary aim of the work was to determine the value of using the MASCOT philosophy for ATE software and the results of this part of the work are described in reference 5. The method adopted to tackle the objective was slightly unusual in that it hinged on the translation of ATLAS into a pseudo bus language rather than a direct compilation into ATE machine code. The pseudo language is in fact comprised of a set of CORAL 66 macro calls which when formed into a CORAL 66 program could be compiled into ATE machine code. The advantage gained being the ability to use commercially available software packages to compile, assemble, link and locate CORAL 66 programs on the target ATE.

The result of earlier research work on the derivation of a flexible Machine Independent Compiler for ATLAS (MICA)⁶ was an ATLAS compiler resident on a PDP 11/40 at RSRE. A block diagram of MICA is shown in figure 1. MICA comprises two major modules, a pre-compiler which performs grammatical analysis and a post compiler which allocates ATE resources to ATLAS statements and generates ATE machine code. Because the post compiler was written with machine independence in mind it is a complex and large program and because many of the resource allocation problems tackled by the post compiler are unnecessarily complex it was decided simply to make use of the MICA pre-compiler for this research work. The MICA pre-compiler was therefore used to perform the grammatical analysis phase of the ATLAS compilation and output a representation of the ATLAS program, known as ATLAS Intermediate Form (AIF). The AIF was then transported to a MDS, shown in figure 2, and a translation program, written in Pascal, converted the AIF to the bus pseudo language statements and generated a resultant CORAL program. This resultant program once compiled, using a commercially available CORAL 66 compiler, formed what could be regarded as the MASCOT root activity.

4.1 An Outline Description of the Software Aspects of the Research Work

The principles involved in the work are best described by a simple example. The ATLAS program for which is shown below:

```
E000000 BEGIN, ATLAS PROGRAM 'TEST' $  
C $  
05 DECLARE, DECIMAL, 'DUMMY' $  
C $  
10 DISPLAY, MESSAGE, CONNECT DVM TO BUS, CONNECT RES TO BE  
MEASURED PRESS GO WHEN READY $  
C $  
15 WAIT FOR, MANUAL INTERVENTION $  
C $  
20 MEASURE, (RES), IMPEDANCE, RES MAX 10 KOHM  
CNX HI J1-1 LO J1-2 $  
24 DISPLAY, MESSAGE, MEASURED RESISTANCE= $  
26 DISPLAY, RESULT, 'MEASUREMENT' $
```

30 DISPLAY, MESSAGE, KILOHMS \$
000255 REMOVE, ALL \$
000265 FINISH \$
000270 TERMINATE, ATLAS PROGRAM 'TEST' \$

Following pre-compilation of the ATLAS program, using MICA, a file containing the AIF was generated. A brief description of AIF and a listing of the AIF generated for the above program is given in Appendix A.

The AIF was then translated using a look-up table technique, into the pseudo bus statements and finally into a CORAL program. The equivalent CORAL program of the above example being:

```
'CORAL' TEST
'LIBRARY' "INOUT.EXT"
'LIBRARY' "F1:AC1.EXT"
'LIBRARY' ":F1:KLMACS"
'LIBRARY' ":F1:IEEE01.MAC"
'ABSOLUTE' (
'PROCEDURE' TTYOUT/'HEX'(FC9F) ('VALUE''BYTE');
);
'EXTERNAL' ('LABEL' AC1);
'BEGIN'
'LIBRARY' ":F1:IEEE01.SET"
'BEGIN'
'LIBRARY' ":F1:IEEE01.PRC"
DECLARE;

'COMMENT'*****START OF TEST PROCEDURE*****;

CLEAR;
DISPLAY("*C*LCONNECT DVM TO BUS.CONNECT RES TO BE MEASURED*C*L*
*PRESS GO WHEN READY*C*L");

WAIT FOR MANUAL INTERVENTION;
SWITCH DVM(1); (CONNECT DVM TO UUT VIA MATRIX)
MEASURE OHMS; (DVM ON AUTORANGE)
DISPLAY("*C*LMEASURED RESISTANCE=");
OUTPUT RESULT;
DISPLAY("KILOHMS*C*L");
REMOVE ALL;
```

```
'COMMENT'*****END OF TEST PROCEDURE*****;  
'END';  
'END'  
'FINISH'
```

Each pseudo statement within the body of the resultant CORAL 66 program is a macro or procedure call, the full expansions of which will not be described in this paper. However, it is generally true to say that they were, by virtue of the bus characteristics, of a simple and repetitive nature in CORAL 66 with assembly code inserts.

4.2 An Outline Description of the Hardware Aspects of the Research Work

A bus controller board, figure 3, was developed for the work and was used on both the MDS and a minimum configuration target ATE, the rear view of which is shown in figure 4. The board, described in reference 5, utilised the INTEL 8291 LSI chip and the INTEL 8292, a pre-programmed microprocessor in its own right. Other experimental hardware includes a BCD instrument to bus conversion unit together with a simple switching matrix. Both these devices employed the Mullard HEF 4738V LSI. Figure 5 shows the bus listener element of the switching matrix.

5 CONCLUSIONS

Past experience has shown that the task of compiling ATLAS into ATE machine code is a large and complex task and there are few examples of success in this field. However the technique as described in this memorandum, virtually an ATLAS to CORAL 66 translation process, could potentially reduce some of the traditional problems such as code generation. It would also seem that the use of a rigorously defined and popular bus, such as the IEEE-488 bus can also bring about a simplified approach. Of course the exercise described in this memorandum was, by necessity, of a limited nature and further work would clearly need to be done to prove the real benefit or otherwise of the scheme.

REFERENCES

- 1 IEEE Standard Digital Interface for Programmable Instrumentation. IEEE Standard 488-1978.
- 2 IEEE/ARINC Standard ATLAS Test Language and ATLAS Syntax. IEEE Standard 416-1976.
- 3 Pascal User Manual and Report by Kathleen Jensen and Niklaus Wirth. Published by Springer-Verlag.
- 4 The Official Handbook of MASCOT. MASCOT Suppliers Association.
- 5 The Advantages of the IEEE-488 bus and the MASCOT philosophy in Automatic Testing. RSRE Memorandum Number 3285. July 1980 by Mr M O'Brien.
- 6 Machine Independent Compiler for ATLAS (MICA)
RSRE Memorandum 3167 by D P Taylor.

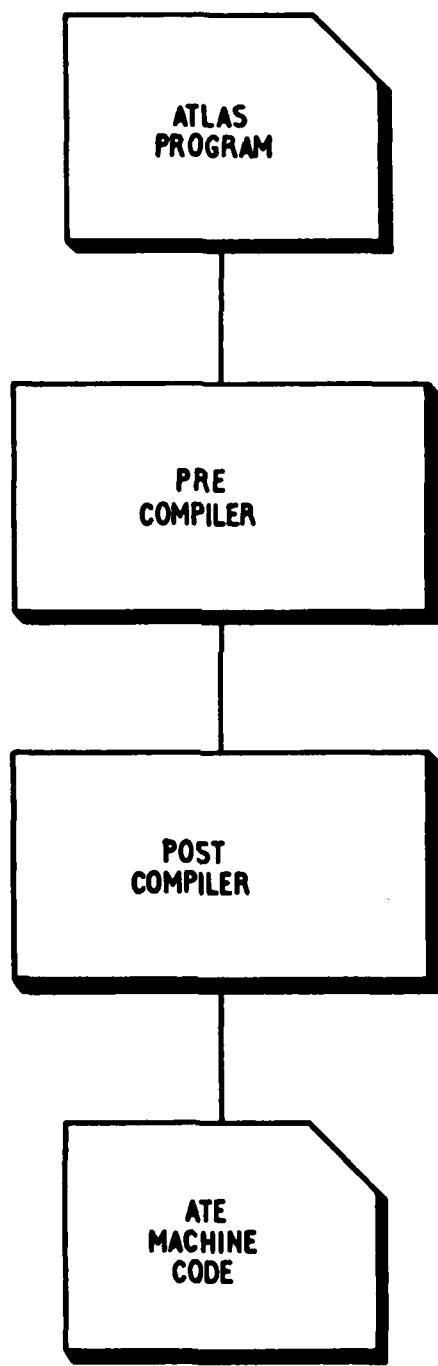


FIG.1 MICA BLOCK DIAGRAM



FIG. 2 MICROPROCESSOR DEVELOPMENT SYSTEM (MDS)

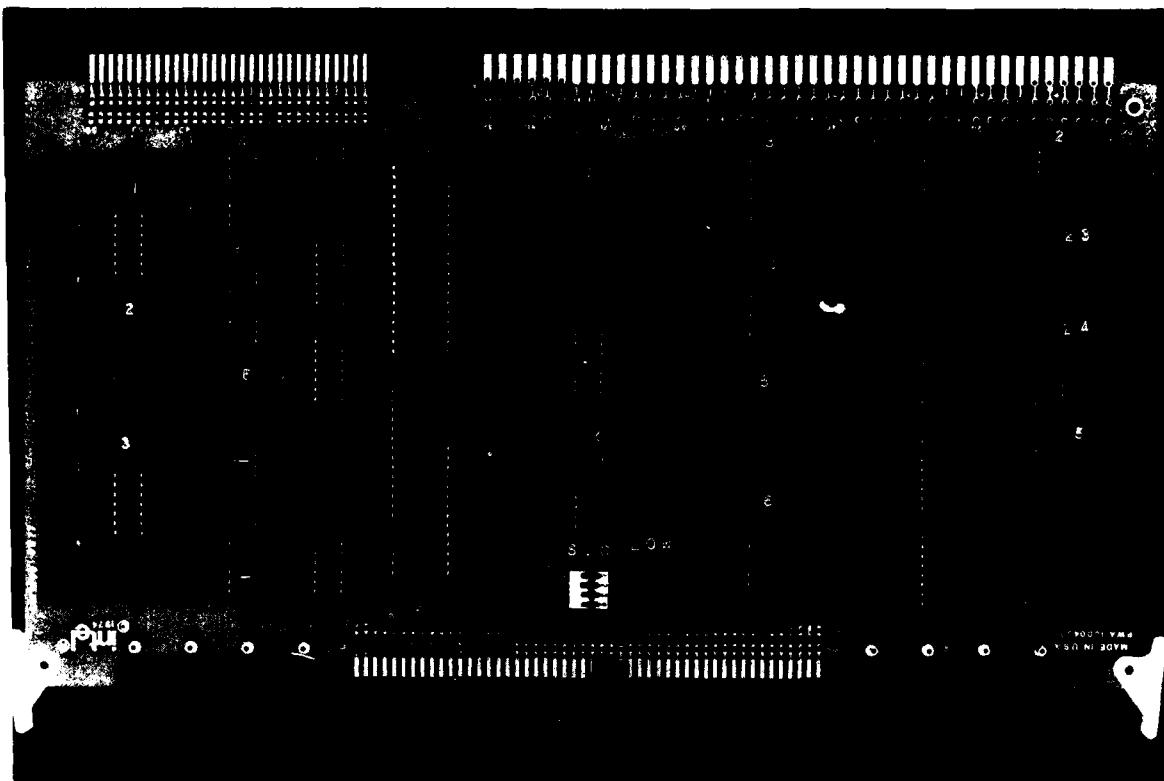


FIG.3 BUS CONTROLLER BOARD

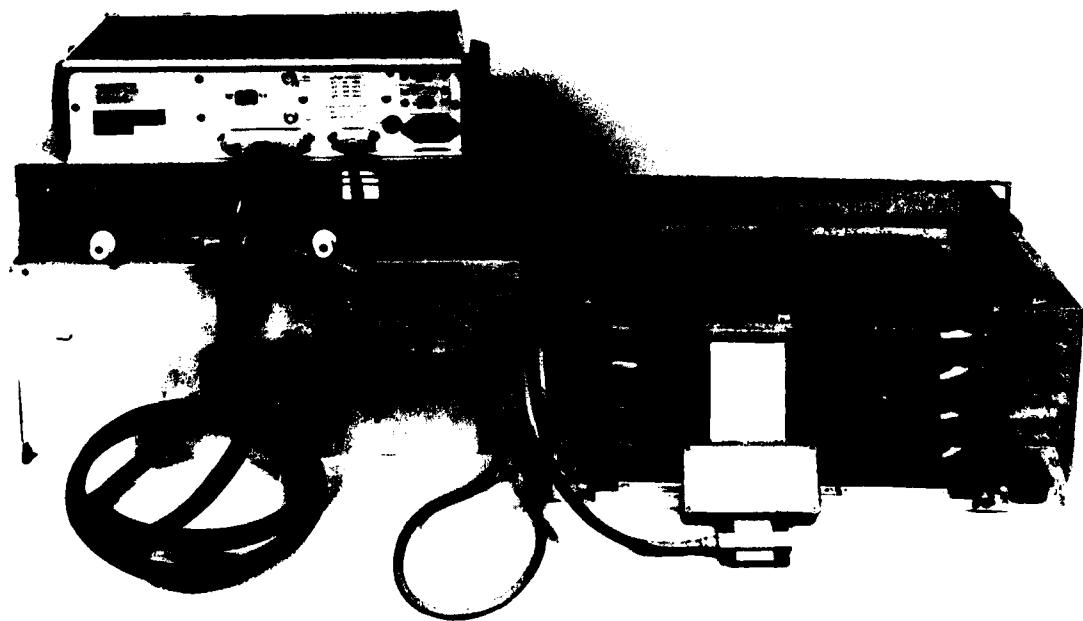


FIG.4 REAR VIEW OF THE TARGET ATE

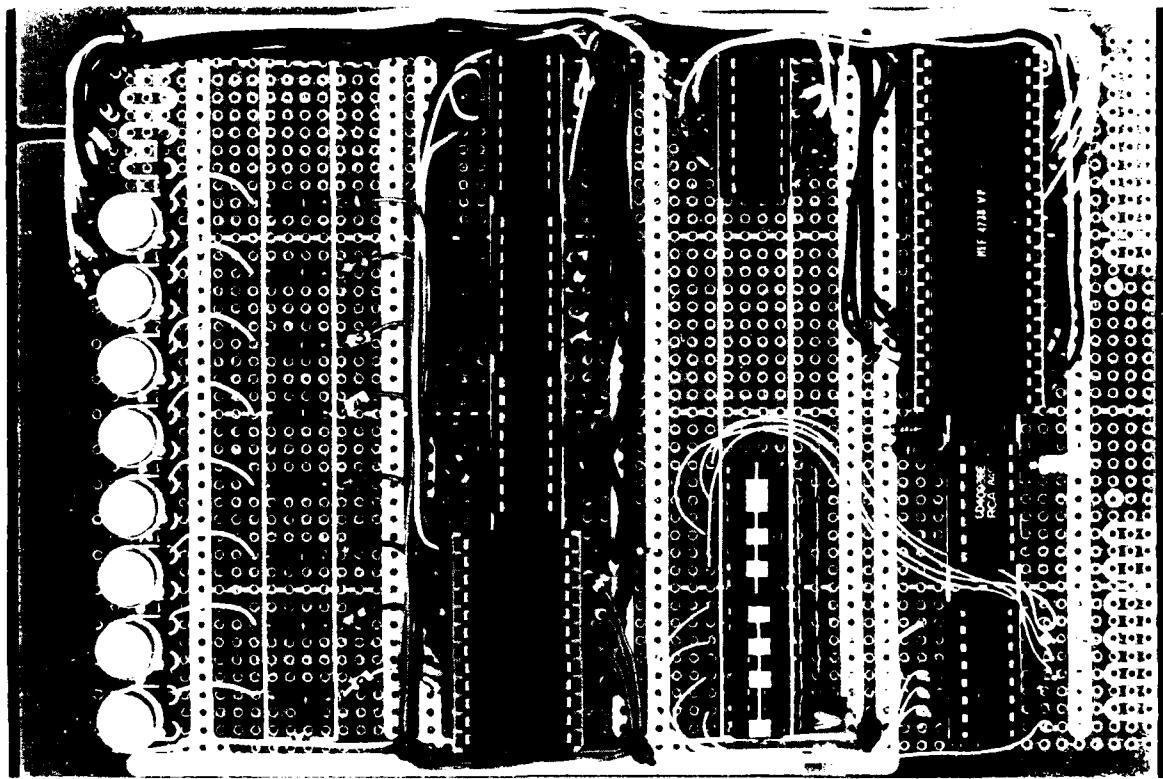


FIG. 5 LISTENER ELEMENT OF THE SWITCHING MATRIX

APPENDIX A

A.1 Format of ATLAS Intermediate Form (AIF)

The AIF is in ASCII and is a reverse polish representation of the source ATLAS program and is formed from two basic types, operators and operands. Preceding operands forming the arguments of the following operator. Certain operators do not have associated operands and these are preceded by two blanks whereas operators which have associated operands are preceded by blank then \$.

A.2 An Example of ATLAS Intermediate Form

The AIF for the example ATLAS program given in paragraph 4.1 is given below:

```
0    RK $2D
1    RL $2D
0    RM $2D
0    RQ $2D
0    RS $2D
0    RR $2D
0    T2 $2D
0    RN $2D
0    I2 $2D
0    RO $2D
0    RP $2D
E000000 $0K 000001001 $0J 41 $3B $6L
000010 $0K 000003002 $0J 94
CONNECT@DVM@TO@BUS,CONNECT@RES@TO@BE@MEASURED 000003003 $0J
@@@@@@@@@@@PRESS@GO@WHEN@READY $OE $30 $6L
000015 $0K 000004001 $0J C7 Q1 $30 $6L
000020 $0K 000005002 $0J B9 MB $1G GF MB P4 10 $01 1Z $0A 000005003
$0J $29 $26 $00 $0M $0L Q0 P5 J1-1 $0C $23 P6 J1-2 $0C $23 $40 $6L
000024 $0K 000006001 $0J 94 MEASURED@RESISTANCE=@ $OE $30 $6L
000026 $0K 000007001 $0J 94 X1 $3S $30 $6L
000030 $0K 000008001 $0J 94 KILOHMS@ $OE $30 $6L
000255 $0K 000009001 $0J B8 $4R $40 $6L
000265 $0K 000010001 $0J AB $6L
000270 $0K 000011001 $0J 42 $3B $6L
$76
```

4x3

18400099786
10 An Experimental Implementation of ATLAS Programs on an IEEE-488 Bus
structured ATE
ESU
1190ct 80
140R3PE-MEMO-3284
1800R1C
1900R1-77373
2000U
2000S

Sam M. Weller